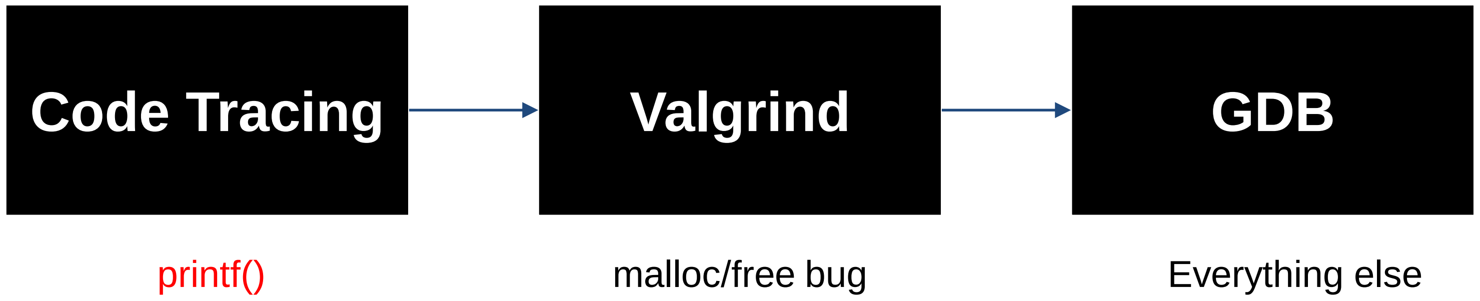
Introduction to GDB and Debugging

What is GDB?

- GNU Debugger
- Supports multiple languages including C
- Inspect program execution at any point
- Helps debug errors such as segmentation faults
- Documentation: http://sourceware.org/gdb/current/onlinedocs/gdb/

Tools for Debugging



Compiling with Debugging

- Add -g option to enable debugging support
- Example: gcc –g program.c

Starting GDB

- Command: gdb ./filename
- Or load later using: (gdb) file prog1.x
- Starts interactive debugging shell

GDB Tips

- Interactive shell with history and autocomplete
- Use help command for assistance
- Command format: (gdb) help [command]

Running the Program

- Command: (gdb) run
- Runs program inside debugger
- Shows useful crash information (line number, variables, etc.)

Handling Bugs

- Set breakpoints to stop program at specific points
- Step through code line by line
- Analyze errors interactively

Setting Breakpoints

- Command: (gdb) break file1.c:6
- Stops execution at specified line
- Supports multiple breakpoints
- Conditional breakpoints available (example: (gdb) break file1.c:6 if i >= ARRAYSIZE)

Continue and Step

- Run until breakpoint: (gdb) continue
- Execute line by line: (gdb) step
- Skip subroutine details: (gdb) next

Inspecting Variables

- Command: (gdb) print var
- Print value of variables during execution
- Print hex value: (gdb) print/x var

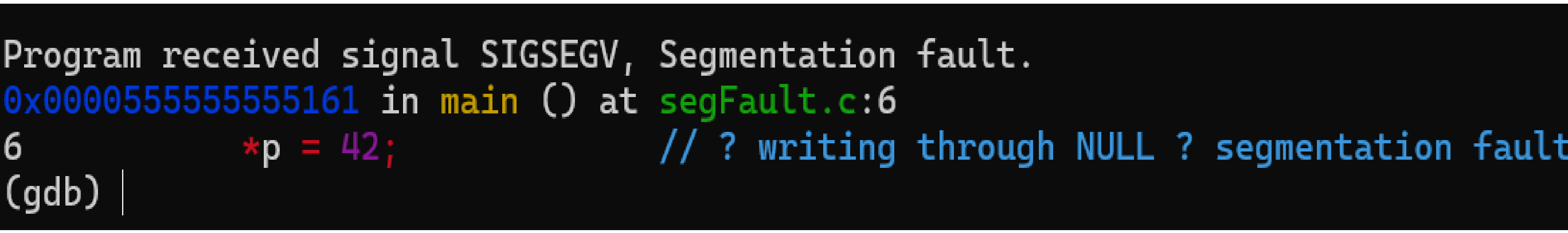
Watchpoints

- Interrupt program when variable changes
- Command: (gdb) watch var
- Scope determines which variable is watched

Example-segmentation Fault

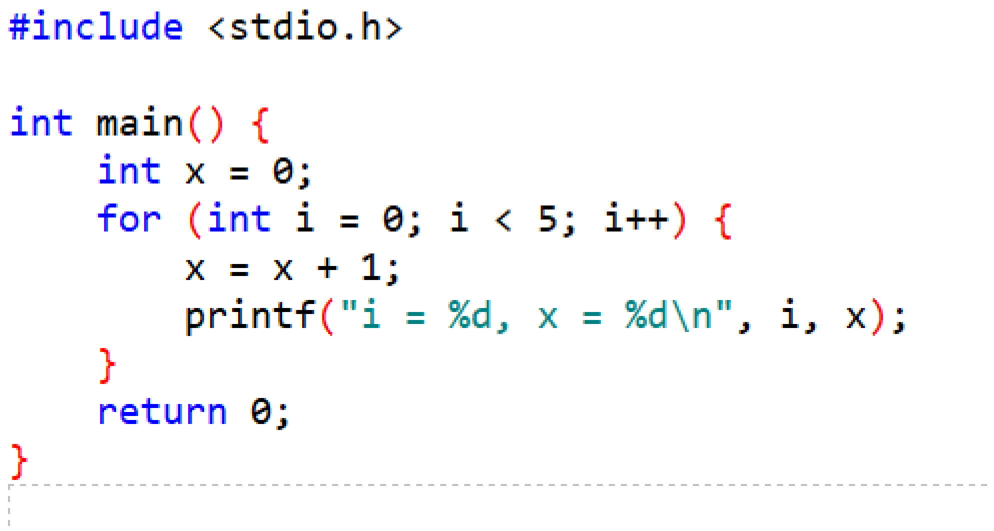
```c
// segfault_demo.c
#include <stdio.h>

int main() {
    int *p = NULL;
    *p = 42;
    printf("%d\n", *p);
    return 0;
}
```



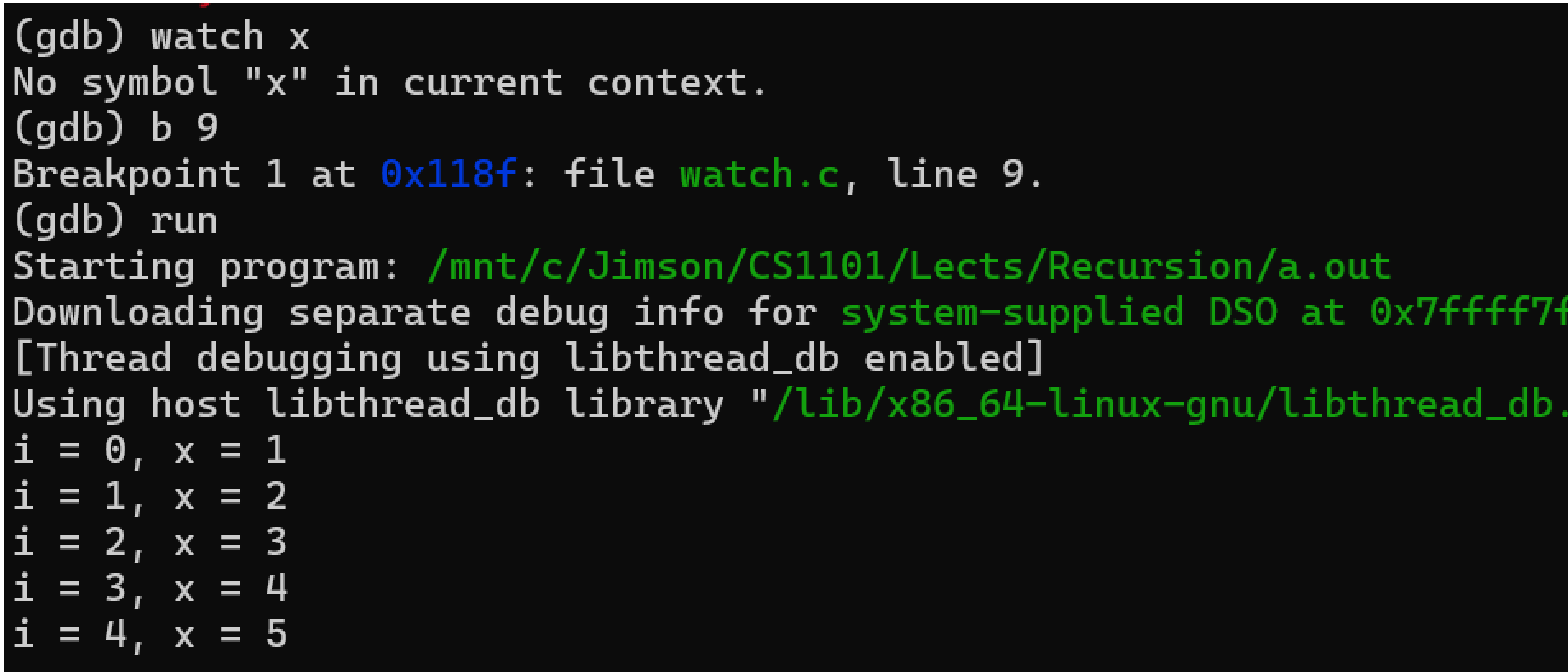
```
Program received signal SIGSEGV, Segmentation fault.
0x0000555555555161 in main () at segFault.c:6
6           *p = 42;            // ? writing through NULL ? segmentation fault
(gdb) 
```

Watchpoints



```c
#include <stdio.h>

int main() {
    int x = 0;
    for (int i = 0; i < 5; i++) {
        x = x + 1;
        printf("i = %d, x = %d\n", i, x);
    }
    return 0;
}
```



```
(gdb) watch x
No symbol "x" in current context.
(gdb) b 9
Breakpoint 1 at 0x118f: file watch.c, line 9.
(gdb) run
Starting program: /mnt/c/Jimson/CS1101/Lects/Recursion/a.out
Downloading separate debug info for system-supplied DSO at 0x7ffff7f
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.
i = 0, x = 1
i = 1, x = 2
i = 2, x = 3
i = 3, x = 4
i = 4, x = 5
```

Extra Commands

- backtrace – Show stack trace
- finish – Run until current function ends
- delete – Remove breakpoints
- info breakpoints – Show breakpoints information